Introduction

Joe Sabado
Assoc. Director of Information Systems and Software Development
Student Information Systems and Technology
Sabado-j@sa.ucsb.edu

Justin Mead
Application Developer
Student Information Systems and Technology
Mead-j@sa.ucsb.edu
Agenda

- General Overview of Mobile space
- Mobile Web Presence in Higher Education
- Implementation Strategies
  - Business, technical, end-user considerations
  - Comparison of native apps, mobile web, hybrid apps
- Design Strategies/Development Techniques
  - Device Detection
  - Adaptation Techniques
- Testing
  - Desktop vs Real Device Testing
- Demo
  - SA Professional Development Conference (PDC) mobile website using UCLA Mobile Web Framework
  - HTML 5
- Discussion
Purpose/Background

Background
❖ We are enthusiasts who volunteered to research mobile web and share our findings, still a lot to learn
❖ A my.sa.ucsb.edu mobile web prototype was presented to Division of Student Affairs Managers in May 2010 and due to higher priorities (SIS conversion project) project was postponed
❖ Some changes since prototype
  ❖ Expressed interests from some students
  ❖ UCLA Mobile Web Framework
  ❖ Gartner predictions of tech trends – increased adoption of mobile, social media and cloud computing in the next few years
  ❖ Hybrid application development more mature

Purpose
❖ Introduce business/technical considerations
❖ Share design/development techniques/resources
❖ Start conversation on how to move forward with mobile web as a decentralized campus
Background – MyUCSB Prototype

Presented to Division of Student Affairs Managers – May 2010

- Proposal for mobile web presented by Chad Mandala, Goodspeed Intern
- Features
  - Student Quarter Schedule / Finals
  - Schedule of Classes
  - Calendar of Events
  - OSL Campus Orgs
  - Athletics
  - Emergency Info
  - Alerts
  - Directory
- Student Affairs priority was SIS conversion so project postponed
General Overview of Mobile space
Mobile Browsers Overview

StatCounter Global Stats
Top 9 Mobile Browsers in the United States from Mar 10 to Feb 11

http://gs.statcounter.com/
Mobile vs. Desktop Overview

StatCounter Global Stats
Mobile vs. Desktop in the United States from Mar 10 to Feb 11

http://gs.statcounter.com/
By 2013, mobile phones will overtake PCs as the most common Web access device worldwide.

http://www.gartner.com/it/page.jsp?id=1278413
### General Overview of US Mobile Web Space

#### USA Mobile Web Overview - September 2010

<table>
<thead>
<tr>
<th>Devices</th>
<th>WIFI Capable</th>
<th>Touchscreen</th>
<th>Full Keyboard</th>
<th>Feature-Phones</th>
<th>Smart-Phones</th>
<th>Experience-Phones</th>
<th>Non-Phones</th>
</tr>
</thead>
<tbody>
<tr>
<td>1385</td>
<td>71.4%</td>
<td>60.9%</td>
<td>41.1%</td>
<td>18.4%</td>
<td>30.1%</td>
<td>37.7%</td>
<td>12.8%</td>
</tr>
</tbody>
</table>

#### Vendors

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Devices</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>35.5%</td>
<td></td>
</tr>
<tr>
<td>BlackBerry</td>
<td>22.0%</td>
<td></td>
</tr>
<tr>
<td>Samsung</td>
<td>13.9%</td>
<td></td>
</tr>
<tr>
<td>Motorola</td>
<td>7.2%</td>
<td></td>
</tr>
<tr>
<td>HTC</td>
<td>6.3%</td>
<td></td>
</tr>
</tbody>
</table>

#### Operating Systems

<table>
<thead>
<tr>
<th>Operating Systems</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone OS</td>
<td>35.5%</td>
</tr>
<tr>
<td>Proprietary OS</td>
<td>22.9%</td>
</tr>
<tr>
<td>BlackBerry OS</td>
<td>22.0%</td>
</tr>
<tr>
<td>Android OS</td>
<td>13.1%</td>
</tr>
<tr>
<td>Windows Mobile OS</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

#### USA Country Networks

<table>
<thead>
<tr>
<th>Network</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T, USA</td>
<td>24.6%</td>
</tr>
<tr>
<td>Verizon, USA</td>
<td>17.7%</td>
</tr>
<tr>
<td>Sprint PCS, USA</td>
<td>13.2%</td>
</tr>
<tr>
<td>T-Mobile, USA</td>
<td>10.0%</td>
</tr>
<tr>
<td>Comcast, USA</td>
<td>5.2%</td>
</tr>
<tr>
<td>Metro PCS, USA</td>
<td>4.1%</td>
</tr>
</tbody>
</table>

Source: [http://percentmobile.com](http://percentmobile.com)
Types of Devices

Feature Phones
- They normally come with their own operating system.
- All feature phones do not support third party software. If they do then they might be running on BREW or JAVA.

Smartphones
- Run on the third party operating systems like Windows Mobile, Android, Symbian, Blackberry, iOS operating systems.

Tablets
- Wireless, portable personal computer with a touch screen interface. The tablet form factor is typically smaller than a notebook computer but larger than a smart phone.

Definitions

- **Mobile Web**
  - Resides on a web server accessed via web browsers
  - Written in html, css, javascript
  - Limited access to device hardware capabilities

- **Native Apps**
  - Applications installed on the device, either pre-installed or downloaded from online application stores like Apple’s itunes and Android Market
  - Developed using language native to the platform (objective C on iOS, java on Android, .net on Windows mobile 7)
  - More robust and can operate without internet access
  - Access to device hardware/software capabilities (accelerometer, camera, file system ...)

- **Hybrid Apps**
  - Developed using html, css, javascript but extends access to device hardware using javascript and custom APIs
  - Leverages web development skill sets
  - Available via online application stores
Mobile Web In Higher Education
Higher Ed Technical Survey

Out of **1,789** schools - only **160** or **9%** of schools have an institutional mobile website.

Sample Higher Ed Sites

- Santa Clara University
  - [homepage](#)
  - [course catalog](#)
- West Virginia
  - [OSP Framework (based on MIT)](#)
- Laundry machine availability
  - [University of Northern Iowa](#)
- Computer workstation checker
  - [Temple Tech Center](#)
- North Carolina State University
  - Libraries
- Purdue
  - [Mobile web in the classroom](#)
- Higher Ed mobile web directory

Frameworks/Organizations
- Molly Project
- iMobileU
- [UCLA Mobile Web Framework](#)
UC Mobile Web Sites

Note: Berkeley, San Diego, San Francisco, Davis, Irvine are in the process of evaluating/implementing UCLA Mobile Web Framework.

- UC Berkeley Analysis
- UCSD Presentation to Student Systems Advisory Group
- UCSD Mobile Web Recommendation
- UCLA MWF Information
• What is it?

The UCLA Mobile Web Framework is a framework pilot intended to facilitate the development of [a] robust, feature-rich, cross-platform mobile web presence. Focusing on mobile web standards, semantic markup and device agnosticism, the framework enables developers to create mobile-styled web applications with relative ease of use.

http://mwf.ucla.edu/docs/documentation

• Features of the Framework:

• Stylesheets: The MWF serves CSS tailored to the requesting device

• JavaScript: The MWF serves JavaScript tailored to the requesting device

• Body entities: The MWF provides definitions of “specialized” elements.

• Public: You can use it – immediately!
### User Agents

**Site Name:** my.sa.ucsb.edu  
**Date Range:** 1/1/2011 to 3/21/2011

<table>
<thead>
<tr>
<th>Rank</th>
<th>User Agent</th>
<th>Browser</th>
<th>Version</th>
<th>Platform</th>
<th>Visits</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Mozilla/5.0 (iPhone, U, CPU Phone OS 4_2_1 like Mac OS X; en-us)</td>
<td>Safari</td>
<td>653.13.5</td>
<td>Mac OS X</td>
<td>24,762</td>
</tr>
<tr>
<td>22</td>
<td>AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148</td>
<td>Safari</td>
<td>653.13.5</td>
<td>Mac OS X</td>
<td>15,101</td>
</tr>
<tr>
<td>26</td>
<td>Mozilla/5.0 (iPhone; U; CPU Phone OS 4_1 like Mac OS X; en-us)</td>
<td>Safari</td>
<td>653.11.7</td>
<td>Mac OS X</td>
<td>11,856</td>
</tr>
<tr>
<td>57</td>
<td>AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148</td>
<td>Safari</td>
<td>653.13.5</td>
<td>Mac OS X</td>
<td>5,612</td>
</tr>
<tr>
<td>65</td>
<td>AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8C148</td>
<td>Safari</td>
<td>653.13.5</td>
<td>Mac OS X</td>
<td>4,770</td>
</tr>
</tbody>
</table>

**Other Items (24,799) | 591,345**

**Total(s) | 1,948,556**

**Average(s) | 78**

---

**Note:** Assumption about low mobile device usage is that site is hard to use in current design/structure and as noted by a student, some features are non-functional on Blackberry.
<table>
<thead>
<tr>
<th>Total Page Requests</th>
<th>iOS</th>
<th>Android</th>
<th>Symbian</th>
<th>iPhones</th>
<th>iPads</th>
<th>iPods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul-10</td>
<td>106,446</td>
<td>1,616</td>
<td>187</td>
<td>13</td>
<td>1094</td>
<td>252</td>
</tr>
<tr>
<td>Aug-10</td>
<td>131,422</td>
<td>3,772</td>
<td>276</td>
<td>11</td>
<td>2,852</td>
<td>328</td>
</tr>
<tr>
<td>Sep-10</td>
<td>141,731</td>
<td>3,789</td>
<td>475</td>
<td>15</td>
<td>2,730</td>
<td>572</td>
</tr>
<tr>
<td>Oct-10</td>
<td>174,356</td>
<td>3,942</td>
<td>476</td>
<td>19</td>
<td>2,876</td>
<td>568</td>
</tr>
<tr>
<td>Nov-10</td>
<td>137,936</td>
<td>3,106</td>
<td>281</td>
<td>11</td>
<td>2,289</td>
<td>392</td>
</tr>
<tr>
<td>Dec-10</td>
<td>121,312</td>
<td>2,831</td>
<td>396</td>
<td>1</td>
<td>1,871</td>
<td>547</td>
</tr>
<tr>
<td>Jan-11</td>
<td>167,805</td>
<td>3,864</td>
<td>637</td>
<td>24</td>
<td>2,403</td>
<td>940</td>
</tr>
<tr>
<td>Feb-11</td>
<td>117,469</td>
<td>3,022</td>
<td>406</td>
<td>4</td>
<td>2095</td>
<td>507</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iPhone 4.0</th>
<th>iPhone 3.0</th>
<th>iPhone 2.0</th>
<th>iPhone 1.0</th>
<th>iPad 4.2</th>
<th>iPad 3.2</th>
<th>iPad 4.0</th>
<th>iPad 3.0</th>
<th>iPad 2.0</th>
<th>iPad 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul-10</td>
<td>432</td>
<td>648</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>252</td>
<td>103</td>
<td>156</td>
<td>17</td>
</tr>
<tr>
<td>Aug-10</td>
<td>2,138</td>
<td>697</td>
<td>132</td>
<td>4</td>
<td>0</td>
<td>328</td>
<td>429</td>
<td>149</td>
<td>13</td>
</tr>
<tr>
<td>Sep-10</td>
<td>1,911</td>
<td>801</td>
<td>11</td>
<td>7</td>
<td>0</td>
<td>572</td>
<td>314</td>
<td>145</td>
<td>11</td>
</tr>
<tr>
<td>Oct-10</td>
<td>2,378</td>
<td>463</td>
<td>22</td>
<td>3</td>
<td>0</td>
<td>566</td>
<td>401</td>
<td>90</td>
<td>7</td>
</tr>
<tr>
<td>Nov-10</td>
<td>1,976</td>
<td>303</td>
<td>10</td>
<td>2</td>
<td>33</td>
<td>399</td>
<td>374</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>Dec-10</td>
<td>1,598</td>
<td>250</td>
<td>19</td>
<td>4</td>
<td>243</td>
<td>304</td>
<td>349</td>
<td>59</td>
<td>5</td>
</tr>
<tr>
<td>Jan-11</td>
<td>2,142</td>
<td>245</td>
<td>12</td>
<td>4</td>
<td>611</td>
<td>329</td>
<td>454</td>
<td>53</td>
<td>11</td>
</tr>
<tr>
<td>Feb-11</td>
<td>1840</td>
<td>234</td>
<td>9</td>
<td>3</td>
<td>362</td>
<td>145</td>
<td>372</td>
<td>42</td>
<td>1</td>
</tr>
</tbody>
</table>

Courtesy of Shane Green – UCSB Institutional Advancement
Mobile OS vs Total Page Requests all OS (Windows, OS X, Unix etc.)

- **Page Requests**
  - Total Page Requests
  - iOS
  - Android
  - Symbian

**Graph Details**
- **X-axis**: Months from Jul-10 to Feb-11
- **Y-axis**: Page Requests (0 to 200,000)

**Graph Notes**
-Courtesy of Shane Green – UCSB Institutional Advancement
Mobile OS Page Requests

- iOS
- Android
- Symbian

Courtesy of Shane Green – UCSB Institutional Advancement
A couple of emails from students:

“Hey Joe, Some kids in my cs48 built a mobile gold app for iPhone and are thinking about doing one for droid. Is this something you would be interested in?”

“Just a trimmed down GOLD interface, I know on my blackberry I can see the main page, but nothing else, none of the tabs work. Furthermore the GOLD (mobile) probably doesn’t need all the functions on the side. It NEEDS the ability to view schedule (both weekly and list), as well as the ability so search/add classes. Probably Grades and Registration info as well.

Also something that would be amazing is if there was a unified UCSB schedule. my gold classes, my work schedule, and CLAS etc all in one area. Right now I have to combine all those myself. It would be great to have those all in one place that I could view mobiley and export it to Google calendar or iCal”
Mobile Strategy - Considerations

Balancing Goals and Constraints

❖ Business
   • How does mobile presence fit into your organization’s goals?
   • How does mobile presence help your organization achieve its goals?

❖ Users
   • User goals (and constraints)
   • How does your audience benefit from a mobile presentation?
   • What task will they accomplish with your mobile content?
   • How will they interact?

❖ Technical/Resources
   • Does your organization have the technical infrastructure to develop and maintain the effort?
   • Does your organization have the necessary skill sets/willing to train staff to acquire skill sets to develop/maintain effort?
   • Are you willing to maintain more than 1 site?

❖ Competitors’ Strategies
   ❖ What are other universities doing now? Does your organization need to provide the same service to keep up?
   ❖ How are they approaching their efforts?

By looking at the considerations above, you can determine:
1) Is there a need to provide mobile presence?
2) What type of mobile presence is your org willing and able to pursue?
Implementation Strategies

How to “mobilize” your website

- Do Nothing
  - Display desktop website on mobile devices
  - Some network providers may transcode site for optimization¹
- Create one web site for desktop, one for mobile
  - Generic mobile site (Default Delivery Context)
  - CMS like Drupal, Joomla, Wordpress provide mobile themes/adaptation methods²
- Create one web site for desktop, several versions of mobile site
  - Use adaptation techniques to accommodate device and browser capabilities
- Create native applications for 1 or more platforms
  - iphone, android, windows mobile
- Create hybrid app(s) – PhoneGap³, Titanium Appcelerator⁴

   Drupal - http://drupal.org/project/mobileplugin
## Mobile Strategy - Considerations

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Web App</th>
<th>Native App</th>
<th>PhoneGap App</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Code once and deploy to multiple platforms</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>2. Leverage common dev skills (html/css/javascript)</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>3. Leverage native features of the device</td>
<td>✗</td>
<td>✔️</td>
<td>✗³</td>
</tr>
<tr>
<td>4. App is optimized for performance &amp; user experience</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>5. Enables branding presence on app stores</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>6. Enables monetization option via app stores</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>7. No need for app store submission and approval</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>8. Build <strong>easily</strong> for multiple device types</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>9. Deploy <strong>easily</strong> to multiple device types</td>
<td>✔️</td>
<td>✗</td>
<td>✗⁹</td>
</tr>
<tr>
<td>10. Offline capable</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Challenges

- **Device Fragmentation**
  - More than 6800 devices – different features/form factors
  - Different Versions of mobile web browsers with varying degrees of capability

- **Device Limitations**
  - Small screen size
  - Lack of windows
  - Navigation
  - Lack of Javascript/cookies
  - Types of pages accessible
  - Speed
  - Broken pages (transcoded by service for compression)
  - Cost of bandwidth

- **Context**
  - What is the environment the user is in? (location, disruptions, etc)
  - Why are they using mobile?

1. [http://deviceatlas.com/explorer# /filter///1///1](http://deviceatlas.com/explorer# /filter///1///1)
2. [http://www.quirksmode.org/mobile/browsers.html](http://www.quirksmode.org/mobile/browsers.html)
Lowest Common Denominator Approach

- Defines a minimum set of features that a device have to support. In this case, content is developed based on these guidelines.

- In this approach, developers only create a single version of the content that can work fairly well on as many mobile devices as possible.

- The minimum set of features a device is expected to support is called the Default Delivery Context (DDC).

- DDC is now part of the Mobile Web Best practices 1.0 recommendation by the W3C.

  - Usable Screen Width: 120 pixels, minimum
  - Markup Language: XHTML Basic 1.1 [XHTML-Basic] delivered with content type application/xhtml+xml
  - Character Encoding: UTF-8
  - Image Support Format: JPEG, GIF 89a
  - Maximum Total Page Weight: 20 kilobytes.
  - Colors: 256 Colors, minimum.
  - Style Sheet Support: CSS Level 1 [CSS]. In addition, CSS Level 2 [CSS2] @media rule together with the handheld and all media types (see CSS 2 Media Types).
  - HTTP: HTTP/1.0 [HTTP1.0] or more recent [HTTP1.1].
  - Script: No support for client side scripting.

http://www.w3.org/TR/mobile-bp/
Adaptation (Multi-Serving) Approach

- Delivers content based on the capabilities of the mobile device
- This approach is adaptive in the sense that developers adapt content to work within the constraints of the device
- Developers may create multiple versions of the content to work on as many mobile devices as possible
Adaptation Reasons

- Serving a mobile formatted site rather than the desktop site;
- Swapping style sheets to adapt layout and content to the device’s specific HTML, CSS and JavaScript technology capabilities;
- Modifying the amount or nature of content to suit the device capacity. For example, a JavaScript based slide show may be served to a higher-end device, and a simple list served to a lower-end device;
- Serving smaller or larger images to suit screen size, or swap SVG graphics for bitmap images on less capable devices;
- Enhancing functionality on more capable devices by applying progressive enhancements (based on feature and object detection);
- Increasing or decreasing font size, margins and padding to scale the actionable areas on a touch device.
- Serving different sizes and/or formats of media such as video to suit the device codecs or capabilities.

As the graphs above indicate, the vast majority of devices share *just three screen widths*; 128, 240 and 176 pixels—with many of the remaining values; 120, 130, 160, 208 and 220 pixels—not diverging too far from these three core values.

http://mobiforge.com/designing/story/effective-design-multiple-screen-sizes
## Web Layout Best Practices

<table>
<thead>
<tr>
<th>Device Characteristics</th>
<th>Layout Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Large display</strong></td>
<td>• Although many modern browsers support sophisticated zooming, the use of columns is not recommended. If columns are used, keep their height to a minimum as needlessly tall columns can force the user to scroll up and down repeatedly.</td>
</tr>
<tr>
<td>Over 240 pixels wide</td>
<td>• Tabular data can easily be represented on these devices, especially in landscape mode.</td>
</tr>
<tr>
<td>Strong HTML, CSS, JavaScript support.</td>
<td>• Vertical lists are recommended for menus. Short horizontal lists can be used to create 3-4 item stylised menus.</td>
</tr>
<tr>
<td>Some CSS 3 support.</td>
<td>• Only vertical scrolling is recommended.</td>
</tr>
<tr>
<td><strong>Midsize display</strong></td>
<td>• Avoid columns at this screen size.</td>
</tr>
<tr>
<td>240 pixels wide</td>
<td>• Well-tested and optimised grids can be used.</td>
</tr>
<tr>
<td>Strong HTML, CSS, JavaScript support.</td>
<td>• Vertical lists are recommended for menus. Where possible, limit these lists to 8-10 items.</td>
</tr>
<tr>
<td><strong>Small display</strong></td>
<td>• Use tables for tabular data only. Two column tables can be used, but three or more are typically unsuitable due to screen width unless content consists of short words or numerals only.</td>
</tr>
<tr>
<td>Less than 240 pixels wide</td>
<td></td>
</tr>
<tr>
<td>Likely supports XHTML MP rather than HTML.</td>
<td></td>
</tr>
<tr>
<td>Supports basic CSS and scripting.</td>
<td></td>
</tr>
<tr>
<td><strong>Tiny display</strong></td>
<td>• Avoid columns at this screen size.</td>
</tr>
<tr>
<td>128 pixels wide or less</td>
<td>• Small, well-tested and optimised grids can be used.</td>
</tr>
<tr>
<td>Supports XHTML MP or WAP.</td>
<td>• Vertical lists are recommended for menus. Limit these lists to 6-10 items.</td>
</tr>
<tr>
<td>Very basic CSS support.</td>
<td>• On devices less than 176 pixels, data tables are rarely useable.</td>
</tr>
</tbody>
</table>

Define the use cases (for example, find a product price, find a store near you, call us, or perform a search).

Order the use cases by the most frequent for a mobile user. Use your best guess, statistical information, and usability tests to keep this order updated.

Do your best to make every use case successful in no more than three clicks or at a page depth of no more than three.

Define approximately three main sections below the home page. If you need more, you should separate your service into more mobile pages.

Always offer a link to the desktop website.

Reduce the form pages for user input to the minimum.

http://answers.oreilly.com/topic/1859-navigation-tips-for-your-mobile-website/
Mobile Web Development
W3C Mobile Application Best Practices

- Application Data
- Security and Privacy
- User Awareness and control
- Conservative Use of Resources
- User Experience
- Handling Variations in the Delivery Context

http://www.w3.org/TR/2010/REC-mwabp-20101214/
## Adaptation Options

<table>
<thead>
<tr>
<th>Server-Side Adaptations</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td><strong>Disadvantages</strong></td>
</tr>
<tr>
<td>Offers far greater flexibility as it enables developers to serve (and users to download) only the media and content that is best and most relevant for that particular device.</td>
<td>While fairly reliable, server-side detection is not foolproof as it relies a great deal on the accuracy of a variety of data.</td>
</tr>
<tr>
<td>Can be extremely efficient when combined with device- or group-based caching on the server.</td>
<td></td>
</tr>
<tr>
<td>Typically allows easier design-time minification (compression) and optimisation of assets (CSS files, image sprites, scripts, and so on).</td>
<td></td>
</tr>
</tbody>
</table>

### Client-Side Adaptation

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is simple to execute using a combination of JavaScript and CSS.</td>
<td>Because JavaScript and CSS execute locally, (and depending on implementation) the entire page may have to download before the script can process and manipulate the Document Object Model (DOM).</td>
</tr>
<tr>
<td>Client-side detection can provide a useful fallback and can be used to implement last-minute tweaks related to real-time behaviour such as changes in orientation.</td>
<td>There may be a visible lag between the initial loading and the execution of the client-side adaptation.</td>
</tr>
<tr>
<td>On more capable devices, client-side Ajax and CSS can be used to download additional content and enhancements without having to refresh the entire page</td>
<td>Requires JavaScript, media queries or media types, all of which have variable levels of support and are least supported on older devices.</td>
</tr>
</tbody>
</table>

Device/Browser Detection

- **Server Side**
  - Detect User Agent
  - Pass User Agent to Device Libraries to determine device capabilities (UAProfile)
    - Downloaded list of devices
      - WURFL (xml)
      - DeviceAtlas (json)
      - ASP.Net Mobile Device Browser File
  - Server –Based Solutions
    - Movila DetectFree
    - DetectRight

- **Client Side**
  - CSS media type and media query
  - Determine browser capability
    - Modernizr (HTML 5/CSS)
    - EnhanceJS (basic compatibility issue, box model)
    - Custom capability detection (JS)
Mobile Web Testing
Testing Approaches

- Real device testing
  - Use friends
  - Mobile stores (ask for permission)
- Desktop Testing
  - Simulators
    - Work within desktop browser (web-based)
      - iPhone – testiphone, iphoneTester
      - Opera Mini
      - W3C validator
      - Adobe Dreamweaver Device Central
  - Emulators
    - Download and install on computer
      - Android SDK
      - Blackberry
      - iPhone
  - Browser add-ons or plug-ins
    - Firefox
    - Opera (developer tab)
- Services
  - DeviceAnywhere
  - KeyNote
Demo
- Device Detection
  - Server side
  - Client side
- HTML 5
- UCLA Mobile Web Framework Background
- Professional Development Conference Web App
  - Features
  - Using m.ucla.edu framework
Mobile Web Techniques

Technique #1: Detect device attributes on the server and alter content/redirect the user

Example of mobile redirection using VB.NET and DeviceAtlas:

```vbnet
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    'load up the properties
    Dim tree As Hashtable = Api.GetTreeFromFile(Server.MapPath("~/App_Data/20110309.json"))
    Dim props As Hashtable = Api.GetProperties(tree, Request.Headers("User-Agent"))

    'check to see if we want to override mobile redirection
    If Request.QueryString("override") = "true" Then
        Session("override") = True
    ElseIf Request.QueryString("override") = "false" Then
        Session("override") = False
    End If

    'initialize override value if it isn't set
    If Session("override") Is Nothing Then Session("override") = False

    'redirect if we're a mobile device and not overriding
    If CBool(props("mobileDevice")) And Not Session("override") Then
    End If
End Sub
```
**Mobile Web Techniques**

Technique #2: Use CSS Media Queries to deliver feature-dependent CSS

```css
/* for small rule sets */
@media screen and (min-width : 1200px) {
  /* styles for resolutions of 1200px (wide) and above */
}

/* for large rule sets */
@import url( small.css ) screen and ( min-width: 1200px );

<!-- as a link tag -->
<link rel="stylesheet" media="screen and (min-width: 1200px)" href="small.css" />
```

Queryable Media features:

- width
- height
- device-width
- device-height
- orientation
- aspect-ratio
- device-aspect-ratio
- color
- color-index
- monochrome
- resolution
- scan
- grid

[http://www.w3.org/TR/css3-mediaqueries/#media1](http://www.w3.org/TR/css3-mediaqueries/#media1)
Technique #3: Use JavaScript to detect feature support on the client

```javascript
// without a library
function supports_touch() {
  return ('ontouchstart' in window)
}

// with Modernizr
if (Modernizr.touch) {
  // bind to touchstart, touchmove, etc and watch `event.streamId`
} else {
  // bind to normal click, mousemove, etc
}

// with EnhanceJS
enhance({
  loadStyles: [
    'css/enhancements.css',
    { href: 'css/print.css', media: 'print' },
    { href: 'css/ie6.css', iecondition: 6 }
  ],
  loadScripts: [
    'js/jquery.min.js',
    'js/enhancements.js'
  ]
});
```
Mobile Web Techniques

Technique #4: Use Modernizr’s CSS classes to apply rules only when a feature is supported

```javascript
/* Simulated box shadow using borders */
div.somediv {
    border-bottom: 1px solid #666;
    border-right: 1px solid #777;
}

/* rules for browsers that support box-shadow */
.boxshadow div.somediv {
    border: none;
    box-shadow: #666 1px 1px 1px;
    -moz-box-shadow: #666 1px 1px 1px;
    -webkit-box-shadow: #666 1px 1px 1px;
}
```

Modernizr can detect support for:

- @font-face
- Canvas
- Canvas Text
- WebGL
- HTML5 Audio
- HTML5 Audio formats
- HTML5 Video
- HTML5 Video formats
- rgba()
- hsla()
- border-image
- border-radius
- box-shadow
- text-shadow
- Multiple backgrounds
- background-size
- opacity
- CSS Animations
- CSS Columns
- CSS Gradients
- CSS Reflections
- CSS 2D Transforms
- CSS 3D Transforms
- Flexible Box Model
- CSS Transitions
- Geolocation API
- Input Types
- Input Attributes
- localStorage
- sessionStorage
- Web Workers
- applicationCache
- SVG
- Inline SVG
- SVG Clip paths
- SMIL
- Web SQL Database
- IndexedDB
- Web Sockets
- hashchange Event
- History Management
- Drag and Drop
- Cross-window Messaging
- Touch Events
What is it?

The UCLA Mobile Web Framework is a framework pilot intended to facilitate the development of [a] robust, feature-rich, cross-platform mobile web presence. Focusing on mobile web standards, semantic markup and device agnosticism, the framework enables developers to create mobile-styled web applications with relative ease of use.

http://mwf.ucla.edu/docs/documentation

Features of the Framework:

• Stylesheets: The MWF serves CSS tailored to the requesting device

• JavaScript: The MWF serves JavaScript tailored to the requesting device

• Body entities: The MWF provides definitions of “specialized” elements.

• Public: You can use it – immediately!
UCSB Division of Student Affairs Professional Development Conference Mobile Website

- Built using UCLA Mobile Web Framework
- Intended audience are conference attendees at University Center
- Benefit included saving attendees from having to print 17 page pdf/word documents
- Design/Development period greatly minimized because of UCLA MWF

http://www.sa.ucsb.edu/profdev/m/
The MWF <head>

<!-- UCLA MWF meta -->
<meta name="viewport" content="height=device-height,width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=no;" />

<!-- UCLA MWF scripts -->
<script type="application/javascript" src="http://m.ucla.edu/assets/js.php"></script>

<!-- UCLA MWF styles -->
<link rel="stylesheet" href="http://m.ucla.edu/assets/css.php?theme=dropshadows+blue-on-white" type="text/css" />
</head>

The MWF page header and footer:

<body>

<h1 id="header">
<a href="/"
   <img src="/images/uccb-logo.png" width="75" height="35" alt="UCSB" />
   <span>SA Professional Development Conference</span>
</a>

<!-- Your content will go here -->

<div id="footer">

©2011 Regents of the University of California.<br />
<a href="http://www.sa.ucsb.edu/profdev/?override=true">View Full Site</a>

</div>

</body>
An MWF Menu

MWF Form

MWF Content
Demonstration

http://m.meadmiracle.com/profdev/
• Right now, Device Access = Geolocation

• Plans for the future include access to additional device hardware features:
  • Audio/video input from device cameras and microphones
  • Local contact and event data
  • Device orientation and/or accelerometer

• Device Access gives mobile web applications increased potential for the future

• Frameworks like PhoneGap are already providing Device Access via JavaScript

• Some concerns about Device Access:
  • Current implementations are limited-to-non-existent.
  • Different hardware on different devices
  • Device Access relies on the user’s consent
• Where is Geolocation supported?

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Chrome</th>
<th>Opera</th>
<th>iPhone</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.0+</td>
<td>3.5+</td>
<td>5.0+</td>
<td>5.0+</td>
<td>10.6+</td>
<td>3.0+</td>
<td>2.0+</td>
</tr>
</tbody>
</table>

http://diveintohtml5.org/geolocation.html

• The user must explicitly give permission in order to read their location

  “User agents must not send location information to Web sites without the express permission of the user.”

  http://www.w3.org/TR/geolocation-API/#security

• If GPS is not active on the device, Geolocation will fall back on triangulation

• In short, you CAN use Geolocation...if you take the right precautions
A skeleton example of Geolocation

```javascript
// first we check for geolocation support with Modernizr
if (Modernizr.geolocation) {
    // the default options object
    var geo_options = {
        enableHighAccuracy: false,
        timeout: undefined,
        maximumAge: 0
    };
    // calls a position function once
    navigator.geolocation.getCurrentPosition(geo_success, geo_error, geo_options);
    // calls a position function at intervals
    var watcher = navigator.geolocation.watchPosition(geo_success, geo_error, geo_options);
    // stop watching their position
    navigator.geolocation.clearWatch(watcher);
}
```
A Geolocation success callback

```javascript
/*
position = {
  coords: {
    latitude: [decimal degrees of latitude],
    longitude: [decimal degrees of longitude],
    altitude: [meters above the reference ellipsoid],
    accuracy: [in meters],
    altitudeAccuracy: [in meters],
    heading: [degrees clockwise from true north],
    speed: [in meters/second]
  },
  timestamp: [like a Date() object]
},
*/

function geo_success(position) {
}
```
A Geolocation error callback

```javascript
/*
    positionError = {
        code: [an enumerated value],
        error: [a technical error message (not for end users)]
    }
*/

Code Enum:
    PERMISSION_DENIED (1)
    POSITION_UNAVAILABLE (2)
    TIMEOUT (3)
    UNKNOWN_ERROR (0)

function geo_error(positionError) {
}
```
Geolocation

Demonstration

http://php.meaditation.com/MapTests/QRDirections.htm
?to=34.411844,-119.848137,The_UCen
What is it?

*Web Apps can start faster and work even if there is no internet connection, thanks to the HTML5 App Cache, as well as the Local Storage, Indexed DB, and the File API specifications.*

[http://www.w3.org/html/logo/#offline-storage-desc](http://www.w3.org/html/logo/#offline-storage-desc)

- **Application Cache** – Create a local, offline copy of your web app.
- **Local Storage** – Store key/value pairs locally on the user’s machine.
- **IndexedDB** – A database-like object store, to be implemented in the future.
- **File API** – Access file information *prior* to uploading the file to a remote server.
Local Storage

Can I use it?

HTML5 STORAGE SUPPORT

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>FIREFOX</th>
<th>SAFARI</th>
<th>CHROME</th>
<th>OPERA</th>
<th>IPHONE</th>
<th>ANDROID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8.0+</td>
<td>3.5+</td>
<td>4.0+</td>
<td>4.0+</td>
<td>10.5+</td>
<td>2.0+</td>
<td>2.0+</td>
</tr>
</tbody>
</table>

http://diveintohtml5.org/storage.html

What is it?

- Key/Value pairs, stored as strings (this is important)
- A lot of space – 5 Mb per Origin (similar to host/domain)
- Persistent – data remains until it is cleared programmatically, or by the user

What is it not?

- A cookie – Local Storage data is not transmitted with HTTP Requests
- Expandable – 5 Mb is a hard limit...for now.
Using the Local Storage object

```javascript
// first we check for support with Modernizr
if (Modernizr.localstorage) {
    // store an item
    localStorage.setItem("foo", "bar");
    localStorage["bob"] = "figs";

    // get an item
    var foo = localStorage.getItem("foo");
    var bar = localStorage.key(0);

    // get the length of the storage area
    var length = localStorage.length;

    // remove an item
    localStorage.removeItem("foo");

    // clear the storage area
    localStorage.clear();
}
```
Local Storage

Demonstration

http://diveintohtml5.org/examples/localstorage-halma.html
Can I use it?

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>FIREFOX</th>
<th>SAFARI</th>
<th>CHROME</th>
<th>OPERA</th>
<th>IPHONE</th>
<th>ANDROID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.0+</td>
<td>3.5+</td>
<td>4.0+</td>
<td>5.0+</td>
<td>10.6+</td>
<td>2.1+</td>
<td>2.0+</td>
</tr>
</tbody>
</table>

How does it work?

- Step 1: User loads the page while connected to a network.
- Step 2: Browser recognizes and reads the Cache Manifest.
- Step 3: Browser attempts to create/update the Application Cache.
- Step 4: User loads the page while not connected to a network.

http://diveintohtml5.org/offline.html
Referencing the Cache Manifest

```html
<!DOCTYPE html>
<html lang="en" manifest="myCache.manifest">

The Cache Manifest

```CACHE MANIFEST
# styles
/styles/StyleSheet.css
/styles/jquery.selectbox.css

# scripts
/scripts/jquery-1.5.1.min.js
/scripts/jquery.selectbox-0.1.min.js
/scripts/jquery.validate.js
/scripts/modernizr-custom.min.js

# images
/images/background_graddiv.gif
/images/header_graddiv.jpg
/images/select-icons.png

# other
/AjaxHandler.ashx?action=minorlist
```
Application Cache

Demonstration

http://diveintohtml5.org/examples/offline/halma.html
Social Media Presentation

Social Media Presentation
Monday, April 25, 9:30 am - 12 noon